

EXHIBIT 11

IN THE UNITED STATES PATENT AND TRADEMARK OFFICE

In re the Patent Application of:

John Michael Jensen

Serial No. 09/776,498

Filed: February 5, 2001

For: **METHOD AND SYSTEM TO
FACILITATE FEE BASED
COMMUNICATION**

Examiner: Naresh Vig

Art Unit: 3629

DECLARATION OF JOHN MICHAEL JENSEN UNDER RULE 1.131

EXHIBIT 11

NetBuys CGI help

Cgi-bin scripts: How do I use them?

4.1) OVERVIEW

- 4.1.1) What are cgi-bin scripts?
- 4.1.15 Short List Of CGI Commands
- 4.1.2) where should I put cgi-bin scripts?
- 4.1.3) what is the path of date, mail, perl, ...?
- 4.1.4) Charges

4.2) FEEDBACK FORMS, GUESTBOOK & SEARCH

- 4.2.1) add.html
- 4.2.2) mail.txt
- 4.2.3) How to use cgiemail and formmail
- 4.2.4) How to use the Guestbook
- 4.2.5) How to use Search.cgi

4.3) IMAGEMAPS

4.4) HTML PAGE COUNTERS

- 4.4.1) index.cgi
- 4.4.2) Using index.cgi
- 4.4.3) Charges

4.5) READY TO RUN SCRIPTS & ALL SITE COUNTERS

- 4.5.1) Site Counter using SSI
- 4.5.2) WWWBoard
- 4.5.3) Secure and Unsecure order form Samples
- 4.5.4) Password Protected directory
- 4.5.5) Database and Search engine business accounts Only

4.6) TROUBLESHOOTING

4.1) OVERVIEW

4.1.1) What are cgi-bin scripts?

"cgi" stands for "common gateway interface", a fancy name meaning computer programs running on the web server that can be invoked from a www page at the browser. The "bin" part alludes to the binary executables that result from compiled or assembled programs. It is a bit misleading because cgi's can also be Unix shell scripts or interpreted languages like Perl.

A typical use for cgi is the processing of online forms. When the user fills in the boxes on the form and hits the SUBMIT button, the cgi program specified in the html will be run at the server, and the information in the boxes become available to the program as parameters. The program, being a program, can then do anything the programmer wanted it to do.

"cgiemail", for example, is a canned program written in C that gathers up the contents of the boxes on the form and emails them to a specified destination, then sends a www page confirming the action.

4.1.2) where should I put cgi-bin scripts?

Put your cgi-bin scripts in the www subdirectory named cgi-bin.

If your domain is named company.com, you would then access your scripts as /cgi-company/

4.1.3) what is the path of date, mail, perl ...?

To find the path of the program date, type which date. Do this for any program that you need your cgi-bin script to access.

I see lots of perls, which one should I use?

They should all be the same. /usr/local/bin/perl should work nicely.

4.1.4) Charges

NetBuysm can create and design scripts for you. For very simple ones, we normally charge \$60/script. We can provide a quote on more complicated ones. We provide the following standardized scripts for free: feedback forms, imagemaps, page counter scripts.

4.2) FEEDBACK FORMS AND SIMILAR FORMS

Cgiemail is a program written in the C language that takes the contents of fill-in boxes on a form and emails them to a specified destination. In addition to the form specification in the .html file, a mail specification in a .txt file is required to format the resulting email message.

NetBuys CGI help

we provide cgiemail in the cgi-bin directory of each machine. Thus, it would be called with a URI such as /cgi-bin/cgiemail/mail.txt. Details are provided below.

While there are a number of subsections below this one, they all work together and are meant to be read from start to finish.

4.2.1) add.html

If you've never dealt with HTML forms before, don't worry. They're easy to create and understand. (And if you briefly tried that link and thought, nooooo!!! I don't want to read this!, scroll down a bit. The easy examples are at the bottom. You should not need to follow it at all to understand what will be explained presently.)

The form prompts the user for data which is sent to the server as simple key-value pairs. Each <input> tag specifies a record. The key is given by the name attribute, and the value is given by the value attribute. The type attribute tells the browser what kind of data to expect. Now, try looking at the example.

Please note that the hidden items are used to transmit critical info to cgiemail. They provide the location of the success file, the name of the person the results should be sent to, and the subject of the form. When making your own forms, you would definitely (please!) change the e-mail address in the "required-to" field, and likely the subject in the "subject" field. The first item tells cgiemail what to show the user after successfully completing the form. You can, but don't need to, customize this.

After that come the items that are actually presented to the user. You'll want to use type=text input items with cgiemail; it's a simple tool. The size=60 tells the browser how big to make the box. The name=something is required in each input tag, otherwise the browser wouldn't know how to send the data to the server. The value="" attribute is correct in most cases, unless you want a default value in the form.

Note that if a field begins with required-, cgiemail will require that the user enter a value for this field. This is particularly useful if you want to require a user to submit their e-mail address.

When the user presses the Submit button, the data goes to our machine where cgiemail starts doing something with it. What it does is controlled by ...

4.2.2) Mail.txt

Make sure that you upload mail.txt in ASCII mode.

Failure to upload mail.txt in ASCII mode will generate the message:

Server Error

The server encountered an internal error or misconfiguration and was unable to complete your request.

Now that we have all this data, what do we do with it? Mail it, of course! But for flexibility, cgiemail requires that you create a mail.txt file to show it what to send. (If you didn't want flexibility you'd use a mailto link.) The program will read in this file, perform substitutions, and pass it to the mail system.

First, how does cgiemail know where to find this file? Look at the first line in the example form used in the previous section. The bit after the word cgiemail in the action attribute tells cgiemail where to look. When called from netbuys.com, cgiemail does not know what your domain is because it is installed for everyone. So, the first component of this path must be your domain. After that, it goes to your www directory and looks from there. If you have a domain name, we suggest that you explicitly put it in the action attribute, because the behavior is different: cgiemail does know what your domain is if called for your domain.

Fred.com might want to put the mail.txt file in his www directory. His form would then start off with

```
<form method=post action="http://www.fred.com/cgi-bin/cgiemail/mail.txt">
```

BTW, there's nothing magical about the name mail.txt. Feel free to call it mail1.txt, or form1.mail, or what suits you, as long as the form has the correct name for what you uploaded.

Here's the mail.txt we used with the form presented earlier. It's pretty simple. Note that the first several lines are mail headers. You probably shouldn't

NetBuys CGI help
change that part, or the corresponding parts in your form. In particular, there must be a 'To:' header or the mail won't go anywhere!
what cgiemail does is simply replace every string that looks like [key] with the value the user typed in to the field with name=key. That's all. You can lay out your form as is best for your users, but lay out your mail.txt as is best for you to read. You can even insert gobs of text to help format the output. Only the [key] parts will be replaced by cgiemail.

4.2.3) How to use cgiemail
Actually, if you've read the previous two sections you should know enough to use it. We'll summarize here.

Get a sample add form and sample mail.txt file. If you prefer to edit things on-line, upload them to your www directory.
Replace 'thelist' in the form with your domain.
Edit the form to have the input areas you like.
Edit mail.txt to use that input.
Have your variables start with the name required- if you want to require that the user input these fields. (Like 'required-to' in the example.) Remember to change both the form (.html) and template (mail.txt).
You may alter the success variable in the HTML form to change the message someone receives after filling out the form.
Wait! what we gave you was an example form, not a complete example HTML file. Put the necessary stuff around it to make it that. (Minimally put it between a <body> </body> pair.)
If you were editing off-line, upload the files.
Try it, and play with it.

Cgiemail was developed at MIT and is copyright 1994, 1995.

Formmail.cgi

This is my favorite form-to-mail handler... It is very configurable....lots of features !!!!

Click here for a copy of Formmail's instructions to be mailed to you automatically.

Formmail is a program written in the Perl language that takes the contents of fill-in boxes on a form and emails them to a specified destination. In addition it is also capable of generating its own Confirmation page.

Setting Up the FormMail Script:

=====

The FormMail.pl script does not have to be extensively configured to get it to work. There are only two variables in the perl file which you will need to define along with changing the top line of your script to match the location of your perl interpreter.

Necessary Form Fields:

=====

There is only one form field that you must have in your form, for FormMail to work correctly. This is the recipient field.

Field: recipient

=====

Description: This form field allows you to specify to whom you wish for your form results to be mailed. Most likely you will want to configure this option as a hidden form field with a value equal to that of your e-mail address.

Syntax:

<input type=hidden name="recipient" value="email@your.host.xxx">

Optional Form Fields:

=====

NetBuys CGI help

Field: subject

Description: The subject field will allow you to specify the subject that you wish to appear in the e-mail that is sent to you after this form has been filled out. If you do not have this option turned on, then the script will default to a message subject: WWW Form Submission

Syntax:

If you wish to choose what the subject is:

```
<input type=hidden name="subject" value="Your Subject">
```

To allow the user to choose a subject:

```
<input type=text name="subject">
```

Field: email

Description: This form field will allow the user to specify their return e-mail address. If you want to be able to return e-mail to your user, I strongly suggest that you include this form field and allow them to fill it in. This will be put into the From: field of the message you receive.

Syntax:

```
<input type=text name="email">
```

Field: realname

Description: The realname form field will allow the user to input their real name. This field is useful for identification purposes and will also be put into the From: line of your message header.

Syntax:

```
<input type=text name="realname">
```

Field: redirect

Description: If you wish to redirect the user to a different URL, rather than having them see the default response to the fill-out form, you can use this hidden variable to send them to a pre-made HTML page.

Syntax:

To choose the URL they will end up at:

```
<input type=hidden name="redirect" value="http://your.address/to/file.html">
```

To allow them to specify a URL they wish to travel to once the form is filled out:

```
<input type=text name="redirect">
```

Field: required

Version Added: 1.3

Description: You can now require for certain fields in your form to be filled in before the user can successfully submit the form. Simply place all field names that you want to be mandatory into this field. If the required fields are not filled in, the user will be notified of what they need to fill in, and a link back to the form they just submitted will be provided.

Syntax:

If you want to require that they fill in the email and phone fields in your form, so that you can reach them once you have received the mail, use a syntax like:

```
<input type=hidden name="required" value="email,phone">
```

Field: env_report

Version Added: 1.3

Description: Allows you to have Environment variables included in the e-mail message you receive after a user has filled out your form. Useful if you wish to know what browser they were using, what domain they were coming from or any other attributes associated with environment variables. The following is a short list of valid environment variables that might be useful:

REMOTE_HOST - Sends the hostname making a request.

REMOTE_ADDR - Sends the IP address of the remote host making the request.

REMOTE_USER - If server supports authentication and script is

protected, this is the username they have authenticated as.

This is not usually set.

REMOTE_IDENT - If HTTP server supports RFC 931 identification, then

NetBuys CGI help
this variable will be set to the remote user name
retrieved from the server. *This is not usually set.*
HTTP_USER_AGENT - The browser the client is using to send the request.
General format: software/version library/version

There are others, but these are a few of the most useful.

Syntax:

If you wanted to find the remote host and browser sending the request, you would put the following into your form:

```
<input type=hidden name="env_report" value="REMOTE_HOST,HTTP_USER_AGENT">
```

Field: sort

Version Added: 1.4

Description: This field allows you to choose the order in which you wish for your variables to appear in the e-mail that FormMail generates. You can choose to have the field sorted alphabetically or specify a set order in which you want the fields to appear in your mail message. By leaving this field out, the order will simply default to the order in which the browsers sends the information to the script (which isn't always the exact same order they appeared in the form.) when sorting by a set order of fields, you should include the phrase "order:" as the first part of your value for the sort field, and then follow that with the field names you want to be listed in the e-mail message, separated by commas.

Syntax:

To sort alphabetically:

```
<input type=hidden name="sort" value="alphabetic">
```

To sort by a set field order:

```
<input type=hidden name="sort" value="order:name1,name2,etc...">
```

Field: print_config

Version Added: 1.5

Description: print_config allows you to specify which of the config variables you would like to have printed in your e-mail message. By default, no config fields are printed to your e-mail. This is because the important form fields, like email, subject, etc... are included in the header of the message. However some users have asked for this option so they can have these fields printed in the body of the message. The config fields that you wish to have printed should be in the value attribute of your input tag separated by commas.

Syntax:

If you want to print the email and subject fields in the body of your message, you would place the following form tag:

```
<input type=hidden name="print_config" value="email,subject">
```

Field: title

Version Added: 1.3

Description: This form field allows you to specify the title and header that will appear on the resulting page if you do not specify a redirect URL.

Syntax:

If you wanted a title of 'Feedback Form Results':

```
<input type=hidden name="title" value="Feedback Form Results">
```

Field: return_link_url

Version Added: 1.3

Description: This field allows you to specify a URL that will appear, as return_link_title, on the following report page. This field will not be used if you have the redirect field set, but it is useful if you allow the user to receive the report on the following page, but want to offer them a way to get back to your main page.

Syntax:

```
<input type=hidden name="return_link_url"  
value="http://your.host.xxx/main.html">
```

Field: return_link_title

Version Added: 1.3

NetBuys CGI help

Description: This is the title that will be used to link the user back to the page you specify with return_link_url. The two fields will be shown on the resulting form page as:

return_link_title

Syntax:

<input type=hidden name="return_link_title" value="Back to Main Page">

Field: background Version Added: 1.3 Description: This form field allow you to specify a background image that will appear if you do not have the redirect field set. This image will appear as the background to the form results page.

Syntax: <input type=hidden name="background"

value="http://your.host.xxx/image.gif">

Field: bgcolor

Version Added: 1.3 Description: This form field allow you to specify a bgcolor for the form results page in much the way you specify a background image. This field should not be set if the redirect field is. Syntax: For a background color of white: <input type=hidden name="bgcolor" value="#FFFFFF">

Field: text_color Version Added: 1.3 Description: This field works in the same way as bgcolor, except that it will change the color of your text. Syntax: For a text color of Black: <input type=hidden name="text_color" value="#000000">

Field: link_color Version Added: 1.3 Description: Changes the color of links on the resulting page. Works in the same way as text_color. Should not be defined if redirect is. Syntax: For a link color of Red: <input type=hidden name="link_color" value="#FF0000">

Field: vlink_color Version Added: 1.3 Description: Changes the color of visited links on the resulting page. Works exactly the same as link_color. Should not be set if redirect is. Syntax: For a visited link color of Blue: <input type=hidden name="vlink_color" value="#0000FF">

Field: alink_color Version Added: 1.4 Description: Changes the color of active links on the resulting page. Works exactly the same as link_color. Should not be set if redirect is. Syntax: For a visited link color of Blue: <input type=hidden name="alink_color" value="#0000FF">

4.2.4) How to use Guestbook

If you have an account that includes a preconfigured guestbook, You simply need to use the following URL to access it.

<http://yourdomain.com/Guestbook/guestbook.html>

4.2.5) How to use Search.cgi

Search will look at all your html pages for words you enter, and return all pages on a list with links. This program is completely configured and ready to run, just access it with the following URL

<http://yourdomain.com/cgi-yourdomain/search.cgi>

4.3) IMAGEMAPS

'Imagemap' refers to an interaction between a user's browser and the WWW server which allows clicking on an image to select hyperlinks. (Other implementations have a separate /cgi-bin/imagemap program. our server handles them internally. Use the reference style shown below, not one like /cgi-bin/imagemap/mymap.map.) Each designated portion of the image may select different links.

Images that are to be used as imagemaps must be designated as "ISMAP". A companion .map file specifies the coordinates of the portions of the image that are hyperlinks, and the URL or local reference of the link.

The following HTML:

```
<A HREF="name-of-map-spec-file.map">  
<IMG BORDER SRC="name-of-image.gif" ISMAP></A>
```

NetBuys CGI help

tell the browser that "name-of-image.gif" is to be displayed and used as an imagemap. The "name-of-map-spec-file.map" file contains the coordinates of the clickable areas in the image and specify the hyperlinks associated with each clickable area.

The map specification file MUST have extension ".map". While this example assumes the file is located in your www directory, it may be located anywhere under the www directory. In such a case, the path from the www directory must be specified. If you put it in /home/your-id/www/map-directory, then use

Please use the NCSA format for your .map file.

A file named imagemap.conf does NOT have to edited when using our imagemap program. That complication has been removed from our version of imagemap.

Here is a sample file named fo-abc.map:
default http://oceania.org/weblinks.html

poly http://oceania.org/weblinks.html#A 23,14 44,2 66,14 66,39 45,51 23,39

poly http://oceania.org/weblinks.html#A 67,14 88,2 110,14 110,39 89,51 67,39

poly http://oceania.org/weblinks.html#C 111,14 132,2 154,14 154,39 133,51 111,39

Note that the URL's are full URLs. You can't use relative URLs.

The number pairs specify the X-Y coordinates in pixels of the points of the three hexagons that are the clickable areas in fo-abc.map. Clicks outside of the areas specified get the default link.

4.4) HTML PAGE COUNTERS

4.4.1) index.cgi

index.cgi, will be found in your cgi-bin. which you will use to update a counter on your home page. Put this file in your main web directory, you don't need to put it in a special cgi-bin directory. It must, however, be called index.cgi to have the desired effect.

Upload this file by using rz -a with telnet or in ASCII mode if you are using ftp. Otherwise you may run into unusual problems.

Because it is a program, the file must be executable to work. If you are telnetting in, use chmod +x index.cgi to make it so. If you wish to only use FTP, you'll have to type chmod 755 index.cgi after uploading it to have the same effect.

4.4.2) Using index.cgi

Type chmod +x index.cgi to activate index.cgi. Then create a file named .count with one line that contains the number 0. Then type chmod o+w .count to allow the counter to be updated. (chmod 666 .count if you're stuck with FTP.) The counter and index.cgi are in your main page in this example.

When http://your-domain.com is referenced at the browser, instead of delivering the file index.html as normally expected, the program index.cgi is run instead. index.cgi reads the .count file, increments the number in that file by one, and writes the file back on disk. Then it reads in the index.html file, finds the string "#COUNT" imbedded somewhere in it, changes "#COUNT" to 1 + whatever value if found in .count, and sends the modified index.html to the browser.

To create a counter on a page other than your home page, change the source from index.html to the page that you wish to update. You would want to make another copy of index.cgi as well. So for example, if you wished to count a page called orders.html, you would change the source from index.html to orders.html, change the name from index.cgi to orders.cgi and then call this page from other pages by the name orders.cgi. Change two things within orders.cgi: find the string 'index.html' and change it to 'orders.html', and find the string '.count' and change it to '.countorders'. Both are right near the top of the program. Then make a file .countorders instead of .count, but otherwise as directed above.

4.4.3) Charges

We will charge \$25 per home page counter that we install. Of course, you can easily install your own.

4.5) COMPLETE WEB SITE COUNTERS

4.5.1) Site Counter using SSI

Very simple to use, first the page which you would like to count must have the extension .shtml or .sht rather than .htm or .html
put the following on the page where you want the count file to appear.

```
<!--#exec cgi="/cgi-yourdomain/counters/counter.cgi"-->
```

For Example if your domain is fred.com the call would be

NetBuys CGI help
<!--#exec cgi="/cgi-fred/counters/counter.cgi"-->
That all there is to it, this will create a count file for each page the call is placed on, it will just show up as a number when looking at it from a web page. We can set up a graphical counter on your page if this is to complicated... or you just don't want to mess with it. The graphical counter requires a \$10 set-up fee. Click here if you want us to set up the counter. Counter
4.5.2) WwWBoard

The World Wide web Board is already to run, you can call it from a page with the following URL: <http://yourdomain.com/bbs/>
4.5.3) Secure and Unsecure Orderform calls

When we setup your account we provide a sample secure and unsecure order form. These can be easily modified or changed to suite your individual or business needs. When calling a non secure order form there is nothing different, you would use your standard URL: <http://yourdomain.com/order.html> assuming the order form is in your main directory. To call a form or a document as secure you will be using Online marketing International inc. registration key. What this means is the page must be called thru our secure server, so the URL that points to this must be as follows: <https://chapell.com/domainname/secureorder.html>. You will have to keep both secure and non-secure order for ms on your domain, you will find that the calls for each are different because the directory hierarchy changes for the secure form, graphics and cgi scripts will have to be called in two different fashions for each form. If you follow the samples supplied for formation and redirection, you should have no problems using this successfully.

4.5.4) Password Protected web directory

Placed in your WwW directory, you will find a sub directory called secure. This has been setup with the password file in place, and it cannot be entered from the web without first being prompted for a login and password. You will need a brief knowledge of Telnet in order to build the login file for this directory, the steps are below.

Login to your domain via telnet.

At prompt type pass, this is a script we created to make adding to the file easy

Follow the prompts. enter login and then passwd desired twice.

You may enter as many logins as you like, you may also go back and add when ever you wish.

Just hit enter, when prompted for login to escape program.

4.5.5) Database and search engine

The database program is composed of 5 essential files database.cfg, database, post.htm, search.htm and output.html
database.cfg - this is the configuration file for your database.

You need to make sure your unix permissions are set to chmod 666 for this file to work properly

The first line is the location of the database, which has the default value of our test database.

The second line is what page that it should return to after posting to database, the default value is to return to the post page for another entry.

NetBuys CGI help

The third line is where you will start listing any fields you want to have posted to the database.

Example.(just a simple list)

Name
Address
City
State

You may have as many as you like listed for these fields

These become your input fields and they must be associated with the input statements on your posting page for each of the above, you should have a matching input statement that looks like the following on your post page:
<input = text name = "name" size = "25" > <input = text name = "address" size = "25">

Database - This is nothing more than the actual data being stored after it is posted from your posting page.

Post.htm - This page is used to write to the database from a webpage. You should have an input statement for each field used in the database .cfg file. You may also use Query boxes, + Radio buttons. The formation for this page should be as follows:

<form method post action = "http://your domain.com/cgi-bin/database.cgi">
For better understanding look at the sample files in the database directory on your domain.

Email - You may activate this feature by adding the following.

<input type="hidden" name="email" value="youremailaddress"> when active each time someone posts to the database you will receive a email of the content

Search.htm - this is the page that reads the actual database file based on the criteria you would like to search. A good example of this page at work would be <http://cyberlands.com/mall/> this utilizes many of the features this program offers.

The form action for this page should be as follows:

<form method = post action = "http://your domain.com/cgi-bin/dsearch.cgi"> use the preconfigured example as a template that is in the database directory of your domain.

There are several ways to setup searches for your database.

1. Click on names (see <http://cyberlands.com/mall/>.)

This will allow a nice interface for your customer or client, that eliminates the need for a submit button
Use the following call:

EX:

The items before the ? mark call the script, this should always be the same. The first item after the ? mark is the word you are using for the search, then you will notice a & sign the listing after the & sign is the location of the database file (it's directory) in the above cyberlands.com/mall/ in the default sample set your domain, it would be domainname/database/:

2. The other way to do a search page is through checkboxes, radio button, query lists, and straight type in text. You may search by as many different mediums as you like. Look at <http://winedine.com/nycsrch.htm>

For an example of different criteria being used, there are only two variable names you will use in your search page. Keywords and Keywords-predefined.

a. Keywords would be used for a single text entry search.

b. Keywords-predefined is used for multiple inputs where you are searching several words as our example with [winedine](http://winedine.com).

Output.html

You must make sure the unix permissions are set to chmod 666 for this file to work properly

Use the page example placed on your domain as an example.

NetBuys CGI help

Each field you want printed in the output page is inclosed in [test]
These may be layed out anywhere on the page, this effectively becomes a
printout template for your file.

You may surround theses variables with as much normal text as you would
like to have printed with these records

4.6) TROUBLESHOOTING

(This section will grow as we remember to add what we keep doing every day
anyway.)

"When I activate my CGI program, I get back a page that says 'Server
Misconfigured'".

Log on and go to your www/cgi-bin directory. Try running the program manually
to see what errors you get. Remember that when a CGI program produces error
output, that output is discarded and you get the non-specific page you saw. To
check on it, you'll have to run it yourself.

If you get something like

```
bash: ./search.pl: No such file or directory
```

and you know you typed the name correctly, it's almost certain you uploaded
the script from your PC or Macintosh in binary mode. Perl scripts are ASCII
text and must be transferred in ASCII mode. (So are shell scripts and any
other kind of script that starts with '#' on the first line.) You know it's
this if ./search.pl doesn't work and perl search.pl does. Assuming you don't
have any legitimate carriage returns in the program (doubtful), this will fix
it:

```
mv search.pl search.pl.txt && tr -d '\r' search.pl
```

Then there's this response:

Literal @01m now requires backslash at ./bigones line 16, within string
Many scripts that you'll find widely distributed are still written for Perl,
version 4. Our /usr/bin/perl is version 5, which is 99.9% compatible with
version 4, along with many improvements. You've found the other 0.1%.

In Perl 4, you could get away with a statement like

```
$mailaddress = "joe@schmoe.com";
```

because Perl didn't think the '@' sign was special in a string. Perl 5 does,
so you need to escape it, like so:

```
$mailaddress = "joe\@schmoe.com";
```

You can't just change all the '@' signs to '\@', only the ones in strings. To
speed your search, however, Perl gives you the line number of the problem
line(s) in its error message. (To check for them without accidentally running
the program, if that would be a problem, use perl -c script.) Fix them, and
see if it works.

Tom Christianson (well-known in the Perl world) has written The Idiot's Guide
to Solving Perl CGI problems for the comp.lang.perl.misc newsgroup. Not
everything in there is correct for the NetBuysm setup, but it's useful.
This page maintained by NetBuysm Services

Return To NetBuysm

BACK TO NETBUYS

email NetBuysm.

Any Comments? Please Call: 800-634-9518

Member of the Internet Link Exchange

Copyright © 1997 Netbuys. All rights reserved.

Revised: October 14, 1997.

EXHIBIT 12

IN THE UNITED STATES PATENT AND TRADEMARK OFFICE

In re the Patent Application of:

John Michael Jensen

Serial No. 09/776,498

Filed: February 5, 2001

For: **METHOD AND SYSTEM TO
FACILITATE FEE BASED
COMMUNICATION**

Examiner: Naresh Vig

Art Unit: 3629

DECLARATION OF JOHN MICHAEL JENSEN UNDER RULE 1.131

EXHIBIT 12

formmail

form-to-mail 2.0

A web page form e-mailer that can save to file too.

by Bill Kendrick
New Breed Software
nbs@sonic.net
<http://www.newbreedsoftware.com/>

January 6, 1997 - August 14, 1998

WHAT IT CAN DO:

Use form-to-mail to accept e-mail messages via web pages. Simply create an HTML page with a form, set it's "action" to the "form-to-mail.cgi" CGI, and you're set!

You can also have form-to-mail store the data it receives to an HTML file. You can use this like a simple guestbook (you'd probably prefer to use a real guestbook, like my "gbook" instead, though) or a way to have people add messages to a simple "bulletin board."

You can even set form-to-mail up to save the data into a format which is easily read by a database package! (Just make form-to-mail save its output into a flat, tab-delimited file.)

REQUIREMENTS:

Your webserver must be on a Unix or Unix-style operating system. (For other systems, you will probably have to alter the source code.)

You must have the ability to run "CGI" programs. (Talk to your webmaster and/or sysadmin.)

The webserver must have the ability to access its local "mail" command. (Again, talk to your webmaster and/or sysadmin.)

If you have any problems, contact myself and/or your webmaster and/or sysadmin! :)

INSTALLING:

To create the "form-to-mail.cgi" CGI program, all you need to do is type "make" at a telnet prompt while inside the "form-to-mail" directory. (This directory should be wherever you need it to be. For example, if you are forced to run CGI's in a "cgi-bin" directory, then the "form-to-mail" directory should be there.) (Also, this directory should be in an ambiguous place. Remember that you can use "form-to-mail" in as many ways as you want or need to. You only need to install it once!)

The only option you may need to set before compiling is where your Unix system's "mail" program is stored. "form-to-mail" is set up to talk to "sendmail", which is usually located at "/usr/sbin/sendmail", but it should work almost as well with just "mail". Talk to your webmaster and/or sysadmin for help. Once you find out where it needs

```
formmail
to be, change this line in "form-to-mail.c" and recompile:
#define EMAIL_CMD "/bin/mail"
```

THE E-MAIL'S FORMAT:

When you receive the e-mail message from a filled-out form, all of fields they filled out are displayed in your e-mail like this:

Comments: I really enjoyed your page. I found the cheese search engine very useful for my pizza design class.

Name: Joe User

Text that is too long is automatically word-wrapped to approximately 80 characters by "form-to-mail." TABs are converted into SPACES to help the word-wrap work right, and RETURNS are left as they are.

SETTING THE FORM FIELDS:

For form-to-mail to work, you must have a form! In this form, you'll be using <input>, <select> and/or <textarea> tags.

Your form will contain some information needed by the "form-to-mail" CGI.

* _to_address ... WHO TO?

If you want "form-to-mail" to e-mail the form to an address, you must specify that address in a field named "_to_address".

A typical field where the "_to_address" will be placed in the form looks like this:

```
<input type="hidden" name="_to_address" value="joe_user@place.org">
```

The user won't see anything on the page, but the "form-to-mail" CGI will get the information anyway.

If you want, you can make it a pull-down menu, so people can choose who the e-mail goes to! Like this:

Who do you want the e-mail to go to?

```
<select name="_to_address">
<option>fred@abc.com
<option>john@def.com
</select>
<br>
```

NOTE: Currently, multiple "_to_address"es aren't supported. You can, however, send an e-mail to multiple recipients by putting commas between the addresses. For example:

```
<input type="hidden" name="_to_address"
value="joe_user@place.org,jane_doe@domain.com">
```

* _subject ... SUBJECT

formmail

You can also set the subject of the message, or let the user do it themselves:

```
<input type="hidden" name="_subject" value="Survey Form E-mail">
```

or

```
Subject: <input type="text" name="_subject">
```

The default subject (if you don't enter one) is "form-to-mail message" followed by the address of the form page, in parenthesis.

* _from_address ... WHO FROM?

You can let users enter their e-mail address and the e-mail will look like it's actually from them (not the webserver, which it really is from).

"form-to-mail" does this trick by sending a "Reply-To:..." line in the e-mail's header. Most e-mail programs will use the "Reply-To:..." address instead of the "From:..." address (which "form-to-mail" can't alter).

You should just be able to use your e-mail program's "reply" command to reply directly to the address the user entered.

```
Your e-mail address: <input type="text" name="_from_address">
```

* _reply_html ... THANKS A MILLION part 1

When the user clicks the "submit" button to send the message, the CGI spits out a very simple "Thank you" message on the page, and they must use their browser's "BACK" command to get out of it.

If you want, you can create a "thank you" HTML page which will come up, instead.

```
<input type="hidden" name="_reply_html" value="thanks.html">
```

The simple "Thank you" message appears, but this page also has a <meta> "refresh" tag which causes most browsers to instantly jump to the location you specified.

NOTE: If you need to run "form-to-mail" in a "cgi-bin" directory, you will need to place the reply file outside of there, and then reference it with a full URL:

```
<input type="hidden" name="_reply_html"
value="http://site.com/thanks.html">
```

* _reply_mail / _reply_subject ... THANKS A MILLION part 2

You can ALSO have "form-to-mail" send an e-mail to the sender when they submit the form.

Simply specify the name of a file with "_reply_mail". (It must be in the same directory as "form-to-mail.cgi", since an internal security measure DISALLOWS "/" (forward-slash) in the "reply_mail" field! It also must have ".form-to-mail" in the filename. This is also a security measure.)

formmail

If you want, you can specify a subject with "_reply_subject".
(If you don't, "autoresponse" will be the subject.)

```
<input type="hidden" name="_reply_mail"
value="thanks.form-to-mail.txt">

<input type="hidden" name="_reply_subject"
value="Thanks for the note!">
```

* _need ... DON'T FORGET TO FILL IT OUT

Sometimes, people don't fill out all of the form. You can force them to do it with "form-to-mail!"

Here's an example where the "Your Name" field MUST be filled out for the e-mail to be sent, but the "Your E-mail" field is optional:

```
Your Name: <input type="text" name="Name"> <br>
<input type="hidden" name="_need" value="Name">

Your E-mail: <input type="text" name="E-mail">
```

The field named "need" tells form-to-mail which fields must be filled out for the e-mail to be accepted. (The "value" of the "_need" field is the "name" of the field which must be filled out.)

You can of course use as many "_need" fields as you need. (Be careful using it; and be sure to test your form after you've set it up!)

* _need_email ... TEST FOR A GOOD E-MAIL ADDRESS

There's another directive similar to "_need" called "_need_email". It acts exactly like "_need" except not only does the field it's naming need to be filled out, it also has to be in proper e-mail address form. (ie, simply "Joe123" will not be accepted. "Joe123@aol.com" will be. And spaces aren't allowed (yes folks, some people decide to put spaces in their e-mail addresses!!!))

* _need_blank ... I NEED IT, BUT THEY DON'T NEED TO ENTER ANYTHING

If you want to let a user leave a field blank, but still want "form-to-mail" to show the field, you can tell "form-to-mail" that you "need it, but it can be blank".

```
Enter your age (optional): <input type="text" name="Age"> <br>
<input type="hidden" name="_need_blank" value="Age">
```

This is useful to keep your e-mails looking the same, and to make sure that every record in a saved file (see below) has the same number of fields.

* _need_blank_email ... I NEED E-MAIL, BUT IT CAN BE BLANK TOO

This field is simply a combination of "_need_email" and "_need_blank". It's a field that, if left blank, will appear blank in the e-mail you receive and the file you are having "form-to-mail" save to.

formmail

But if it's filled it, the user must enter a valid-looking e-mail address for it to work.

* _out_file ... SAVE TO FILE

If you want, you can automatically save the data into a file. This file can become an HTML file, a plain text file, or even a TAB-delimited file (good for importing it into a database program).

First, specify the file you'd like the data to go into:

```
<input type="hidden" name="_out_file" value="survey.dat">
```

Be sure that the file is accessible by the webserver. This usually means creating a blank file ("touch survey.dat"), and then making it writeable by the webserver ("chmod 666 survey.dat"). Check with your webmaster and/or sysadmin about how your webserver is setup and what you need to do to create output files.

* _startrecord / _delimiter / _endrecord ... HOW TO SAVE IT? *

By default, files are saved with each field value stored on the same line, with "TAB" characters between each field. This is suitable for a database import. (Note that the fields are saved in order of their field name (alphabetically), so they're always saved in the same order.)

You can specify the "start of record", "field delimiter" and "end of record" characters and strings yourself, if you wish!

Since you can't just put any old character in a form field, there are a few special character sequences you need to know about:

%n - End-of-line. Puts a return-carriage in place of the "%n".

%t - Tab. Puts a TAB in place of the "%t".

%(- <. Since putting a "<" or ">" inside of an <input> tag can be...

%) - >. ...bad, these two sequences do it for you.

%' - ". Quotes can mess up your <input> tag's "value" attribute, too. Use "%'" instead.

%% - %. Without this, you'd have no way of actually putting a "%" in one of these strings!

Here's an example... Say you want an "" tag at the beginning of a record, a "
" tag put between each field and an "<hr>" tag between each record (a record being one submission of the form), you simply tell it this through the form:

```
<input type="hidden" name="_startrecord" value="%(li%)">
<input type="hidden" name="_delimiter" value="%(br%)%n">
<input type="hidden" name="_endrecord" value="%(hr%)%n">
```

* _shownames ... REMIND ME WHAT THESE FIELDS ARE?

In a case like the above (where the output file is pretty much a guestbook signature page!), you'll probably want the names of the fields displayed ("Name:", "E-mail:", "Comments:", etc.)

```
formmail
<input type="hidden" name="_shownames" value="yes">
```

Voila, a simple guestbook! (You probably WOULDN'T want "_shownames" set to "yes" for a tab-delimited database, however!)

NOTE: The characters "TAB" and "RETURN" within a field (i.e., someone typed it in when they were entering their e-mail) are replaced with "SPACES" when saving to the file. (That way, if the user typed "TAB" or "RETURN" in an input field, it wouldn't mess up the file if you were going to use it for a database, for example.)

REFERENCE:

A brief reference is also contained at the top of the "form-to-mail.c" file itself.

CREDITS / CONTACT:

form-to-mail

by Bill Kendrick
January 6, 1997 - August 14, 1998

E-mail: nbs@sonic.net
website: <http://www.newbreedsoftware.com>
Phone: 1-530-759-1019

EXHIBIT 13

IN THE UNITED STATES PATENT AND TRADEMARK OFFICE

In re the Patent Application of:

John Michael Jensen

Serial No. 09/776,498

Filed: February 5, 2001

For: **METHOD AND SYSTEM TO
FACILITATE FEE BASED
COMMUNICATION**

Examiner: Naresh Vig

Art Unit: 3629

Confirmation No. 9282

DECLARATION OF JOHN MICHAEL JENSEN UNDER RULE 1.131

EXHIBIT 13

```

#####
#####
# FormMail                                     Version 1.6
#
# Copyright 1996-1997 Matt Wright mattw@worldwidemart.com
# Created 06/09/95                             Last Modified 05/02/97
#
# Matt's Script Archive, Inc.:   http://www.worldwidemart.com/scripts/
#
#####
####
# If you run into any problems while trying to configure this scripts,
help #
# is available. The steps you should take to get the fastest results,
are: #
#     1) Read this file thoroughly.
#
#     2) Consult the Matt's Script Archive Frequently Asked Questions:
#
#         http://www.worldwidemart.com/scripts/faq/
#
#     3) If you are still having difficulty installing this script,
send #
#         e-mail to: scripts-help@tahoenet.com
#
#         Include any error messages you are receiving and as much
detail #
#         as you can so we can spot your problem. Also include the
variable#
#         configuration block that is located at the top of the script.
#
#
#
# Hopefully we will be able to help you solve your problems. Thank you.
#
#####
# COPYRIGHT NOTICE
#
# Copyright 1995 - 1997 Matthew M. Wright All Rights Reserved.
#
#
#
# FormMail may be used and modified free of charge by anyone so long as
this #
# copyright notice and the comments above remain intact. By using this
#
# code you agree to indemnify Matthew M. Wright from any liability that
#
# might arise from its use.
#
#
#
# Selling the code for this program without prior written consent is
#
# expressly forbidden. In other words, please ask first before you try
and #
# make money off of my program.
#
#
#

```

```
#
# Obtain permission before redistributing this software over the
Internet or #
# in any other medium. In all cases copyright and header must remain
intact #
#####
```

FormMail is a universal WWW form to E-mail gateway. There is only one required form input tag which must be specified in order for this script to work with your existing forms. Other hidden configuration fields can also be used to enhance the operation of FormMail on your site. Version 1.6 of FormMail contains a few minor bug fixes, optimized code and more comments. The biggest change in this version is that by default, form fields are now sorted as they appear in the form. Error pages were also beautified a little and two new configuration fields were created. Read the History for a more complete list of changes.

The script, FormMail.pl, needs to be placed in your server's cgi-bin and the anonymous WWW user must have the ability to read/execute the script. If you do not have access to your server's cgi-bin, yet you can execute cgi scripts, you may want to try adding a .cgi extension to the FormMail.pl, so you could move it to FormMail.cgi.

Setting Up the FormMail Script:

The FormMail.pl script does not have to be extensively configured in order to work. There are only two variables in the perl file which you will need to define along with changing the top line of your script to match the location of you Perl interpreter.

Necessary Variables:

```
-----
$mailprog = '/usr/lib/sendmail';
This variable must define the location to your server's sendmail
program. If this is incorrect, form results will not be mailed to
you.

@referers = ('worldwidemart.com','206.31.72.203');
This array allows you to define the domains that you will allow
forms to reside on and use your FormMail script. If a user tries to
put a form on another server, that is not worldwidemart.com, they
will receive an error message when someone tries to fill out their
form.
By placing worldwidemart.com in the @referers array, this also
allows
www.worldwidemart.com, ftp.worldwidemart.com, any other http address
with worldwidemart.com in it and worldwidemart.com's IP address to
access
this script as well, so no users will be turned away.
```

Your FormMail program is now configured.

Form Configuration:
=====

The action of your form needs to point towards this script (obviously), and the method must be POST or GET in capital letters. Version 1.5 of FormMail offers many new ways to code your form to tailor the resulting HTML page and the way the script performs. Below is a list of form fields you can use and how to implement them.

Necessary Form Fields:
=====

There is only one form field that you must have in your form, for FormMail to work correctly. This is the recipient field.

Field: recipient

Description: This form field allows you to specify to whom you wish for your

form results to be mailed. Most likely you will want to configure this option as a hidden form field with a value

equal

to that of your e-mail address.

Syntax:

<input type=hidden name="recipient"
value="email@your.host.com">

Optional Form Fields:
=====

Field: subject

Description: The subject field will allow you to specify the subject that you

wish to appear in the e-mail that is sent to you after this form has been filled out. If you do not have this option turned on, then the script will default to a message subject: WWW Form Submission

Syntax:

If you wish to choose what the subject is:
<input type=hidden name="subject" value="Your Subject">

To allow the user to choose a subject:
<input type=text name="subject">

Field: email

Description: This form field will allow the user to specify their return e-mail address. If you want to be able to return e-mail to your user, I strongly suggest that you include this form field and allow them to fill it in. This will be put into the From: field of the message you receive. If you want to require an email address with valid syntax, add this field name to the 'required' field.

Syntax: <input type="text" name="email">

Field: realname

Description: The realname form field will allow the user to input their real name. This field is useful for identification purposes and will also be put into the From: line of your message header.

Syntax: <input type="text" name="realname">

Field: redirect

Description: If you wish to redirect the user to a different URL, rather than having them see the default response to the fill-out form, you can use this hidden variable to send them to a pre-made HTML page.

Syntax: To choose the URL they will end up at:
<input type="hidden" name="redirect"
value="http://your.host.com/to/file.html">

To allow them to specify a URL they wish to travel to once the form is filled out:
<input type="text" name="redirect">

Field: required

Version: 1.3 & Up

Description: You can now require for certain fields in your form to be filled in before the user can successfully submit the form.

Simply place all field names that you want to be mandatory into this field. If the required fields are not filled in, the user will be notified of what they need to fill in, and a link back to the form they just submitted will be provided.

To use a customized error page, see 'missing_fields_redirect'

Syntax: If you want to require that they fill in the email and phone fields in your form, so that you can reach them once you have received the mail, use a syntax like:

```
<input type=hidden name="required" value="email,phone">
```

Field: env_report

Version: 1.3 & Up

Description: Allows you to have Environment variables included in the e-mail message you receive after a user has filled out your form. Useful if you wish to know what browser they were using, what domain they were coming from or any other attributes associated with environment variables. The following is a short list of valid environment variables that might be useful:

making	REMOTE_HOST	- Sends the hostname making a request.
	REMOTE_ADDR	- Sends the IP address of the remote host
		the request.
script	REMOTE_USER	- If server supports authentication and
have		is protected, this is the username they
		authenticated as. *This is not usually
set.*		
the	HTTP_USER_AGENT	- The browser the client is using to send
		request.

There are others, but these are a few of the most useful.

For more information on environment variables, see:

http://www.cgi-resources.com/Documentation/Environment_Variables/

Syntax: If you wanted to find the remote host and browser sending the request, you would put the following into your form:

```
<input type=hidden name="env_report" value="REMOTE_HOST, HTTP_USER_AGENT">
```

Field: sort

Version: 1.4 & Up

Description: This field allows you to choose the order in which you wish for your variables to appear in the e-mail that FormMail generates. You can choose to have the field sorted alphabetically or specify a set order in which you want the fields to appear in your mail message. By leaving this

field out, the order will simply default to the order in which the browsers sends the information to the script (which is usually the exact same order as they appeared in the form.) When sorting by a set order of fields, you should include the phrase "order:" as the first part of your value for the sort field, and then follow that with the field names you want to be listed in the e-mail message, separated by commas. Version 1.6 allows a little more flexibility in the listing of ordered fields, in that you can include spaces and line breaks in the field without it messing up the sort. This is helpful when you have many form fields and need to insert a line wrap.

Syntax: To sort alphabetically:

```
<input type=hidden name="sort" value="alphabetic">
```

To sort by a set field order:

```
<input type=hidden name="sort" value="order:name1,name2, name3,etc...">
```

Field: print_config

Version: 1.5 & Up

Description: print_config allows you to specify which of the config variables you would like to have printed in your e-mail

message.
This By default, no config fields are printed to your e-mail.
etc. is because the important form fields, like email, subject,
users are included in the header of the message. However some
printed have asked for this option so they can have these fields
wish to in the body of the message. The config fields that you
tag have printed should be in the value attribute of your input
separated by commas.

Syntax: If you want to print the email and subject fields in the
body of your message, you would place the following form tag:

```
<input type=hidden name="print_config"
value="email,subject">
```

Field: print_blank_fields

Version: 1.6

Description: print_blank_fields allows you to request that all form
fields are printed in the return HTML, regardless of whether or
not they were filled in. FormMail defaults to turning this
off, so that unused form fields aren't e-mailed.

Syntax: If you want to print all blank fields:
<input type=hidden name="print_blank_fields" value="1">

Field: title

Version: 1.3 & Up

Description: This form field allows you to specify the title and header
that will appear on the resulting page if you do not specify a
redirect URL.

Syntax: If you wanted a title of 'Feedback Form Results':
Results">
<input type=hidden name="title" value="Feedback Form

Field: return_link_url

Version: 1.3 & Up

Description: This field allows you to specify a URL that will appear, as return_link_title, on the following report page. This field will not be used if you have the redirect field set, but it is useful if you allow the user to receive the report on the following page, but want to offer them a way to get back to your main page.

Syntax:

```
<input type=hidden name="return_link_url"
value="http://your.host.com/main.html">
```

Field: return_link_title

Version: 1.3 & Up

Description: This is the title that will be used to link the user back to the page you specify with return_link_url. The two fields will be shown on the resulting form page as:

```
<ul>
<li><a href="return_link_url">return_link_title</a>
</ul>
```

Syntax:

```
<input type=hidden name="return_link_title"
value="Back to Main
Page">
```

Field: missing_fields_redirect

Version: 1.6

Description: This form field allows you to specify a URL that users will be redirected to if there are fields listed in the required form field that are not filled in. This is so you can customize an error page instead of displaying the default.

Syntax:

```
<input type=hidden name="missing_fields_redirect"
value="http://your.host.com/error.html">
```


Field: background
Version: 1.3 & Up
Description: This form field allow you to specify a background image that
will appear if you do not have the redirect field set.
This image will appear as the background to the form results page.
Syntax:
`<input type=hidden name="background" value="http://your.host.com/image.gif">`

Field: bgcolor
Version: 1.3 & Up
Description: This form field allow you to specify a bgcolor for the form results page in much the way you specify a background image.
This field should not be set if the redirect field is.
Syntax:
For a background color of White:
`<input type=hidden name="bgcolor" value="#FFFFFF">`

Field: text_color
Version: 1.3 & Up
Description: This field works in the same way as bgcolor, except that it will change the color of your text.
Syntax:
For a text color of Black:
`<input type=hidden name="text_color" value="#000000">`

Field: link_color
Version: 1.3 & Up
Description: Changes the color of links on the resulting page. Works in the same way as text_color. Should not be defined if redirect is.

Syntax:

For a link color of Red:
<input type=hidden name="link_color" value="#FF0000">

Field: vlink_color

Version: 1.3 & Up

Description: Changes the color of visited links on the resulting page.
Works

redirect exactly the same as link_color. Should not be set if
is.

Syntax:

For a visited link color of Blue:
<input type=hidden name="vlink_color" value="#0000FF">

Field: alink_color

Version: 1.4 & Up

Description: Changes the color of active links on the resulting page.
Works

redirect exactly the same as link_color. Should not be set if
is.

Syntax:

For a visited link color of Blue:
<input type=hidden name="alink_color" value="#0000FF">

Any other form fields that appear in your script will be mailed back to you and displayed on the resulting page if you do not have the redirect field set. There is no limit as to how many other form fields you can use with this form, except the limits imposed by browsers and your server.

Some of the possible uses of this script are:

- 1) You want to have a form that will be mailed to you, but aren't sure how to write the CGI script for it.
- 2) You are the webmaster of your site and want to allow users to use forms, but not to have their own cgi-bin directories, which can cause security risks to your system. You can set this script up and then allow all users to run off of it.

3) Want to have one script to parse all of your html forms and mail them to you.

History:

Version 1.0	06/11/95	-	This script was created.
Version 1.1	08/03/95	-	A major hole in the script which allowed users to run commands under your server's uid as disabled, thanks to Paul Phillips, who noticed the error.
		-	The ability to redirect the user to a specified HTML file after they filled out a form was added.
Version 1.2	09/23/95	-	If the form field is one of the required or optional 'special' fields, such as redirect, recipient, subject, email, realname, etc... the script will not print these fields to either your mail message or to the user's screen when they are returned to a generic form response.
Version 1.3	01/21/96	-	It helps you so that things do not get duplicated.
		-	Much needed update finally completed
		-	Added form fields: env_report, bgcolor, background, link_color, vlink_color, title, text_color, return_link_title, return_link_url and required.
		-	Security hole, which allowed any user on any system to bum off of your FormMail script, has been plugged up with the \$referers variable.
		-	Report style in return html and e-mail touched up a bit.
Version 1.4	01/23/96	-	Added options: sort, alink_color
		-	Fixed a few bugs from Version 1.3, namely the fact that the link_colors weren't working well.
		-	FormMail now supports both the GET and POST methods.
Version 1.5	02/05/96	-	Sorting of Fields in E-Mail Response Fixed.
		-	print config option added.
Version 1.6	05/02/97	-	Sorting of fields by default was fixed to now sort in the order the fields are passed to FormMail from the web browser, which is usually the same order as they appear in the HTML

form.

- The sort order: directive, env_report and print_config parsing routines were made to better compensate for line breaks and extra spaces in input for ease of use.
- Redirect error causing the redirect option incorrectly work with https (secure was fixed.
- Input of a '0' in a regular form field now recognized as input and sent back to user.
- Output of non-filled in form fields

to

servers)

suppressed.

if

or

1.

route

form to

respect

- E-mail addresses checked for correct syntax designated a required field.
- Fields only printed if they contain a value if the print_blank_fields option is set to
- missing_fields_redirect added so you can users who don't completely fill out the a pre-made HTML page.
- Parts of code optimized, especially in to the way config variables are handled.

Matt Wright - mattw@worldwidemart.com -
<http://www.worldwidemart.com/scripts/>